

2. Lista de Requisitos

2.1 Funcionais

- **Criação de Personagens:**
 - Cadastro de personagens com atributos personalizados (força, destreza, constituição, inteligência, sabedoria, carisma).
 - Escolha de raças e classes.
 - Inventário de itens, habilidades e feitiços.
 - Registro de histórico de aventuras e progresso.
- **Gestão de Campanhas:**
 - Criação e configuração de campanhas (mestre, jogadores, cenário, regras específicas).
 - Sistema de mapas e localizações.
 - Controle de quests e objetivos.
- **Sistema de Combate:**
 - Gestão de turnos e iniciativa.
 - Cálculo automático de ataques, defesas e dano.
 - Registro de condições e efeitos (veneno, paralisia, etc.).
- **Banco de Dados de Itens e Habilidades:**
 - Inclusão e categorização de itens (armas, armaduras, poções, etc.).
 - Sistema para criar e personalizar habilidades e feitiços.
- **Gerenciamento de Sessões:**
 - Planejamento e agendamento de sessões de jogo.
 - Histórico de sessões passadas (notas, eventos principais).
- **Interação e Comunicação:**
 - Chat integrado para comunicação entre jogadores e mestre.
 - Sistema de anotações e lembretes.
- **Visualização de Estatísticas:**
 - Relatórios e gráficos sobre progresso de personagens, desempenho em batalhas, etc.

2.2 Não Funcionais

- **Usabilidade:**
 - Interface amigável e intuitiva.

- Suporte a múltiplos idiomas.
- **Segurança:**
 - Autenticação de usuários.
 - Proteção de dados sensíveis.
- **Performance:**
 - Tempo de resposta rápido.
 - Escalabilidade para suportar múltiplas campanhas e usuários simultaneamente.
- **Portabilidade:**
 - Disponível em plataforma web e mobile.
 - Compatível com os principais navegadores.

3. Planejamento do Software

3.1 Arquitetura do Sistema

- **Frontend:**
 - Framework: React ou Vue.js.
 - Biblioteca de UI: Material-UI ou Bootstrap.
- **Backend:**
 - Linguagem: Node.js ou Python (Django/Flask).
 - Banco de Dados: PostgreSQL ou MongoDB.
 - API RESTful para comunicação entre frontend e backend.
- **Infraestrutura:**
 - Deploy: Docker para contêineres.
 - Hospedagem: AWS ou Heroku.
 - Sistema de controle de versão: Git (GitHub/GitLab).

3.2 Etapas de Desenvolvimento

- **Etapa 1: Definição e Planejamento**
 - Reuniões com stakeholders para definir requisitos detalhados.
 - Criação do backlog do projeto.
- **Etapa 2: Design do Sistema**
 - Design da arquitetura do sistema.
 - Criação de wireframes e mockups da interface.

- **Etapa 3: Desenvolvimento do Backend**
 - Configuração do ambiente de desenvolvimento.
 - Implementação de APIs para cadastro de personagens, gestão de campanhas, sistema de combate, etc.
 - Configuração do banco de dados.
- **Etapa 4: Desenvolvimento do Frontend**
 - Implementação da interface do usuário com funcionalidades básicas.
 - Integração com APIs do backend.
- **Etapa 5: Testes e Validação**
 - Testes unitários e de integração.
 - Testes de usabilidade.
 - Correção de bugs e refinamentos.
- **Etapa 6: Deploy e Monitoramento**
 - Deploy da aplicação em ambiente de produção.
 - Monitoramento de performance e erros.
- **Etapa 7: Manutenção e Atualizações**
 - Feedback contínuo dos usuários.
 - Implementação de novas funcionalidades e melhorias.

4. Cronograma

Fase	Duração Estimada
Definição e Planejamento	2 semanas
Design do Sistema	3 semanas
Desenvolvimento do Backend	6 semanas
Desenvolvimento do Frontend	6 semanas
Testes e Validação	4 semanas
Deploy e Monitoramento	2 semanas
Manutenção e Atualizações	Contínuo

5. Equipe do Projeto

- **Gerente de Projeto:** Responsável por coordenar o projeto, garantir a comunicação entre as partes e manter o cronograma.

- **Desenvolvedores Backend:** Focados na criação de APIs, lógica do sistema e banco de dados.
- **Desenvolvedores Frontend:** Responsáveis pela interface do usuário e integração com o backend.
- **Designer UX/UI:** Criará os wireframes e mockups, além de garantir uma boa experiência do usuário.
- **Testadores:** Realizarão testes unitários, de integração e de usabilidade.
- **DevOps:** Gerenciará o deploy e a infraestrutura.

6. Ferramentas Utilizadas

- **Gestão de Projetos:** Jira ou Trello.
- **Design:** Figma ou Adobe XD.
- **Controle de Versão:** GitHub ou GitLab.
- **Comunicação:** Slack ou Microsoft Teams.
- **Documentação:** Confluence ou Notion.

7. Riscos e Mitigações

- **Escopo não Definido:** Realizar reuniões detalhadas com stakeholders.
- **Problemas de Integração:** Garantir comunicação constante entre equipes de frontend e backend.
- **Atrasos no Cronograma:** Manter um buffer de tempo e realizar revisões regulares do progresso.